METHODOLOGIES FOR DISCRETE-EVENT MODELLING AND SIMULATION

(SYSC-5104)

CELLULAR AUTOMATA FOR SALT AND PEPPER NOISE FILTERING

Assignment 2

Yogirajsinh Pravinsinh Rathod

Student#:101088921

yogirajsinhpravinsin@cmail.carleton.ca

November 23, 2017

PART- I INTRODUCTION

Image Processing and Cellular Automata

In various applications of image processing, cellular automata can be applied significantly. i.e. Object detection, edge detection, image enhancement and noise removal techniques.

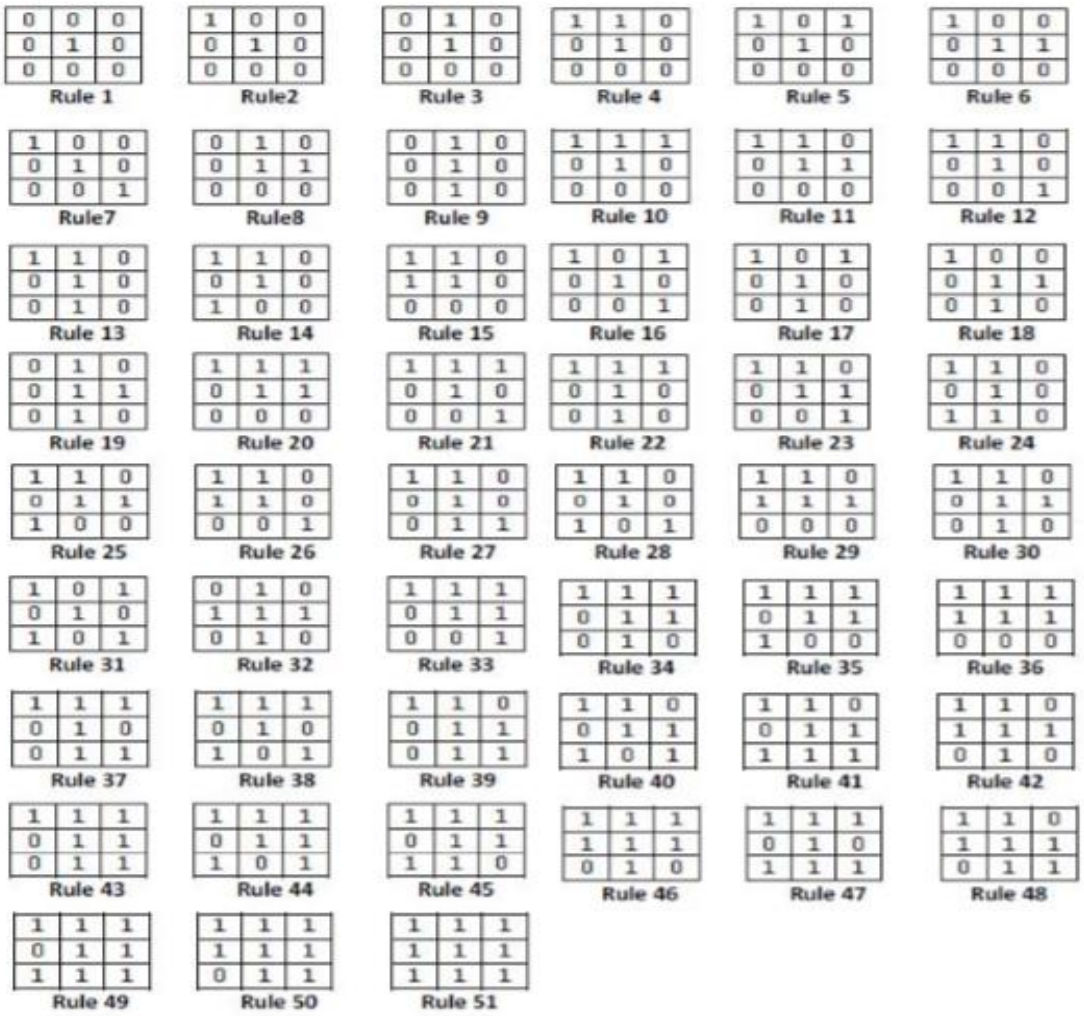
Salt and Pepper Noise Removal

Noise removal in an image is one of the fundamental processes of image processing. In this assignment, I am planning to use cellular automata for noise filtering in a two-dimensional image.

In this experiment, binary images are considered which means cells have two states i.e. white or black.

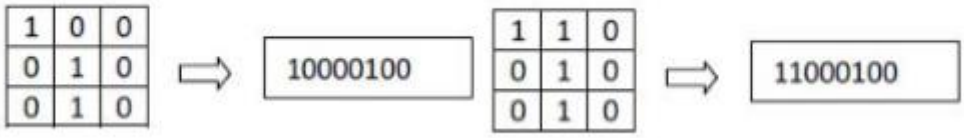
Here, Moore neighborhood is considered, and transition rules are solely applied to non-boundary cells.

As the Moore neighborhood is considered, 2^8 rules are possible which are reduced to 51 by considering 45-degree rotational symmetry and bilateral reflection.



These rules for the central white pixels and same number of rules are found for the central black pixel in similar manner.

The neighborhood pattern is encoded in form of string as shown in figure for central white pixel and the same string is compared with the encoded neighbors of the input image. If both are the same, then the central pixel is inverted.



A sample for converting rule set into string

For central black pixel, the neighborhood pattern is inverted and then compared with the rules.

Step 1 the initial set is empty.

Step 2 at each iteration, all rules are considered for addition to the rule set and only the rule giving the maximum score is added the resulting rule set. This process is repeated until no improvements in score are gained by adding rules.

Step 3 each rule in rule set found in step 2 is removed to find the rule whose removal provides the resulting rule set with the improved value of objective function.

Step 4 if removal of the rule causes the better score of the objective function then it is discarded from the rule set and again next rule is tried for the deletion and process go to step 3.

Otherwise, the process goes to step 2 for the addition of new rule to the rule set.

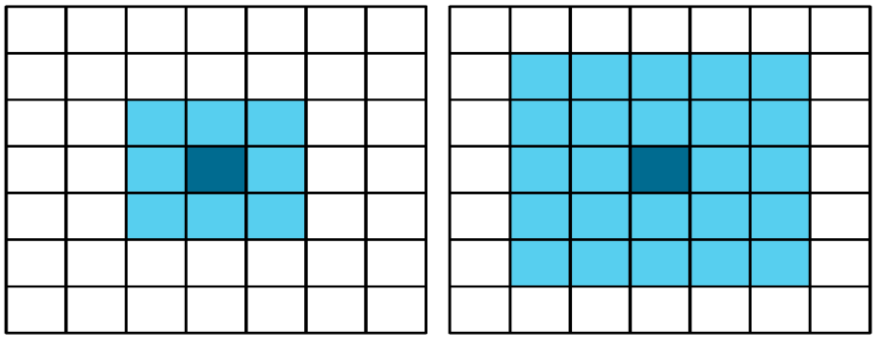
MODIFICATION IN PART I

I have assumed to analyze the image in binary format in CELL DEVS. After analyzing this, I come to conclusion that noise removal is very important to do in Image Processing. I simulated one fundamental aspect of image processing i.e. Salt and pepper Noise Removal Filter. Idea of this filter is to detect and remove the noise from the image.

NEIGHBORHOOD

For image processing, it is a general practice to assume neighborhood of 3x3, either working in cellular automata or not. So, I have assumed to consider neighborhood of 3x3 for my assignment.

MOORE NEIGHBORHOOD



ALGORITHM

Generally, the computation involved in noise detection and removal filter is same, and they only differ in thresholds. Good edge detection techniques also included noise removal property in them like Sobel Edge detector or Laplacian Edge Detector.

As I am working on Salt and Pepper noise removal, so I assumed that if in the neighbor hood, there are more than 2 pixels which have same value, then it is a valid image data.

(0,0) = 1 and trueCount > 2

If a one-valued pixel has a neighborhood of less than 2 pixels, then it is an invalid image data (noise).

(0,0) = 1 and trueCount < 2

If a zero-valued pixel has a neighborhood of greater than 7 pixels, then it is an invalid image data (noise).

(0,0) = 0 and trueCount > 7

Combining rules of noise detection and noise removal filter, I come up with following rules.

rule : 0 100 { (0,0) = 1 and trueCount < 2 }

rule : 1 100 { (0,0) = 1 and trueCount > 2 }

rule : 1 100 { (0,0) = 0 and trueCount > 7 }

rule : 0 100 { t }

PART- II

FORMAL SPECIFICATION FOR SALT AND PEPPER NOISE FILTERING MODEL

The following are the formal specification for the Cell-DEVS Salt and Pepper Noise Filtering model:

CD = < X, Y, I, S, θ, N, d, δint, δext, τ, λ, D >

X = Y = {0 , 1}

I = <η, μ, Px, Px>

η = 9, the neighbourhood size (This varies with each simulation to show the effect of different neighbourhoods. Other neighbourhood sizes will be of 5 and 25.)

μ = {φ}

Px = {φ}

Px = {φ}

S = { Real numbers} where each unique number represents a unique colour

B = wrapped

N = { (-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1) } for η = 9

m=16; n=16;

d = 100 ms

δint : {φ}

δext : {φ}

τ : *For every cell in the image {*

If the cell is the different state than the majority of it s neighbours

Choose the majority cells' state value

If the cell is the same state as the majority of it s neighbours

keep the state of the cell unchanged

}

λ : {φ}

D : 100 ms

RESULTS

After building model, I have tested it and visualized the results in Celldevs Simulation Viewer, and in CD++ Modeler one by one.

I take Smirking Face Emoji, which has Salt and Pepper noise, as an input and then tested the model. Salt and Pepper noise is removed from the background of the image, and also from the object in the image.

RESULTS IN CELLDEVS SIMULATION VIEWER

In the results, Yellow nodes shows Disinfecting nodes, Green nodes shows unprotected nodes and grey nodes shows disinfected nodes.

At 00:00:00:000

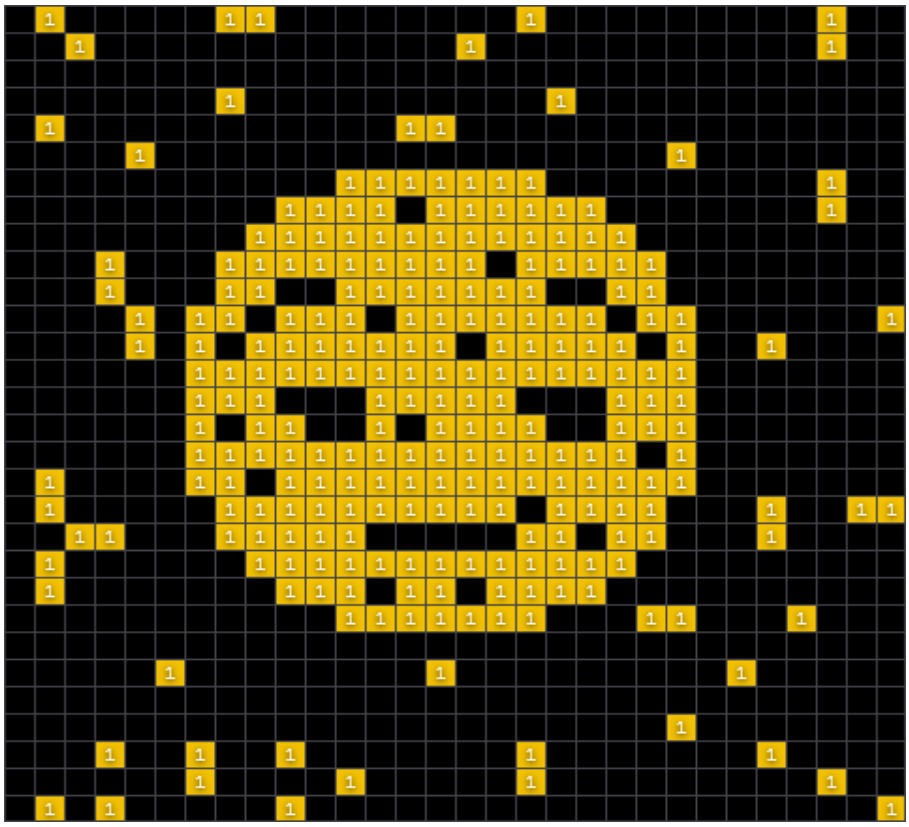


Figure 1 Smirking Emoji Image with Salt and Pepper Noise

At 00:00:00:100

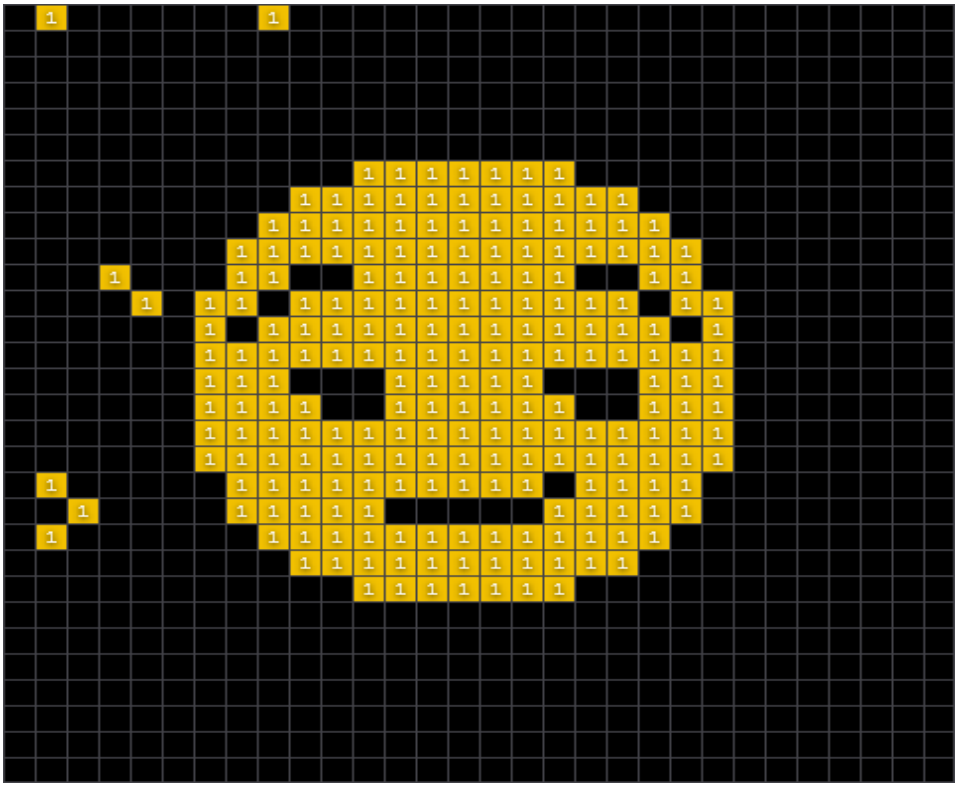


Figure 2 Noise Removed from the Object (Smirking Emoji) completely

At 00:00:00:200

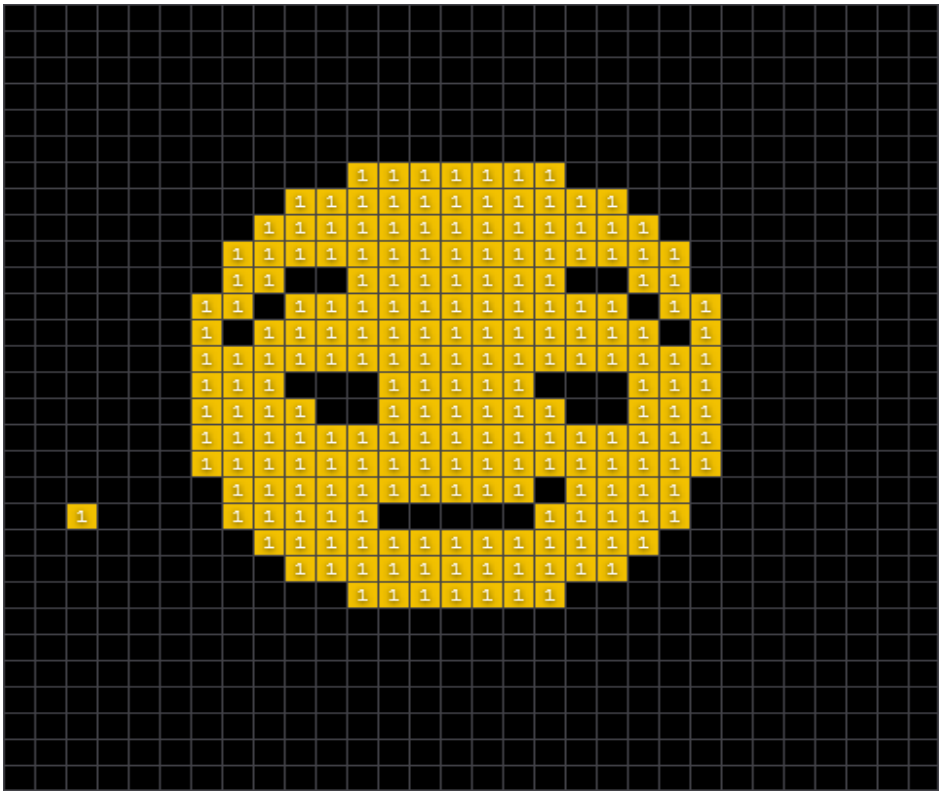


Figure 3 Noise is decreased in the background

At 00:00:00:300

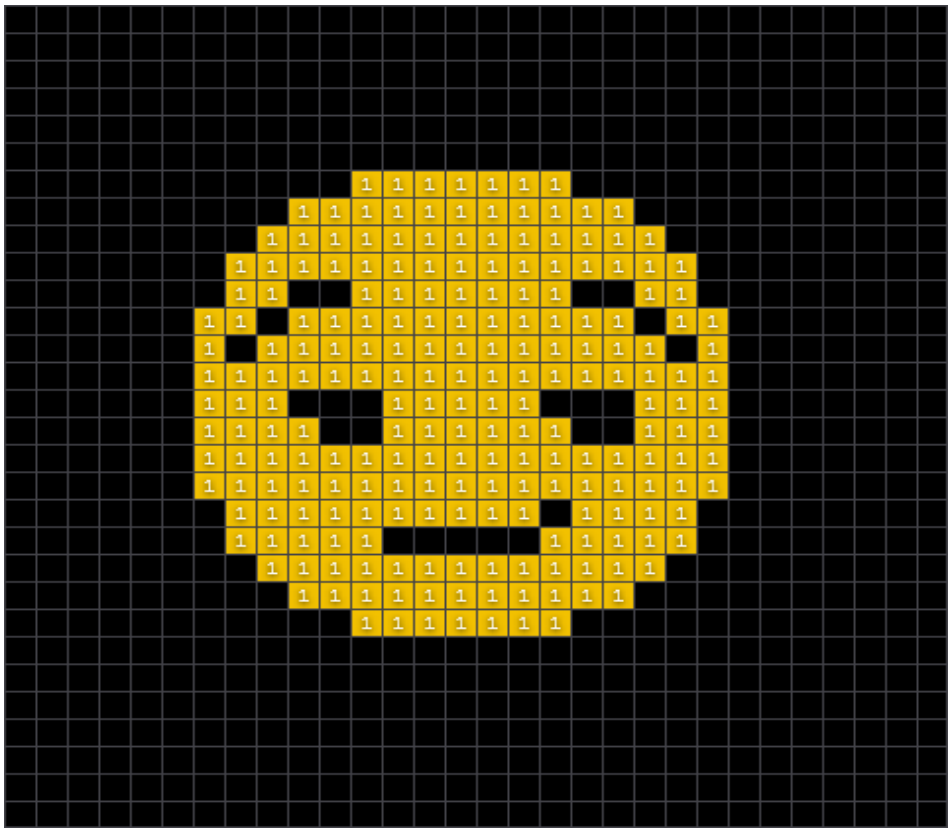


Figure 4 Noiseless Image as a Final Output

RESULTS IN CD++ MODELER

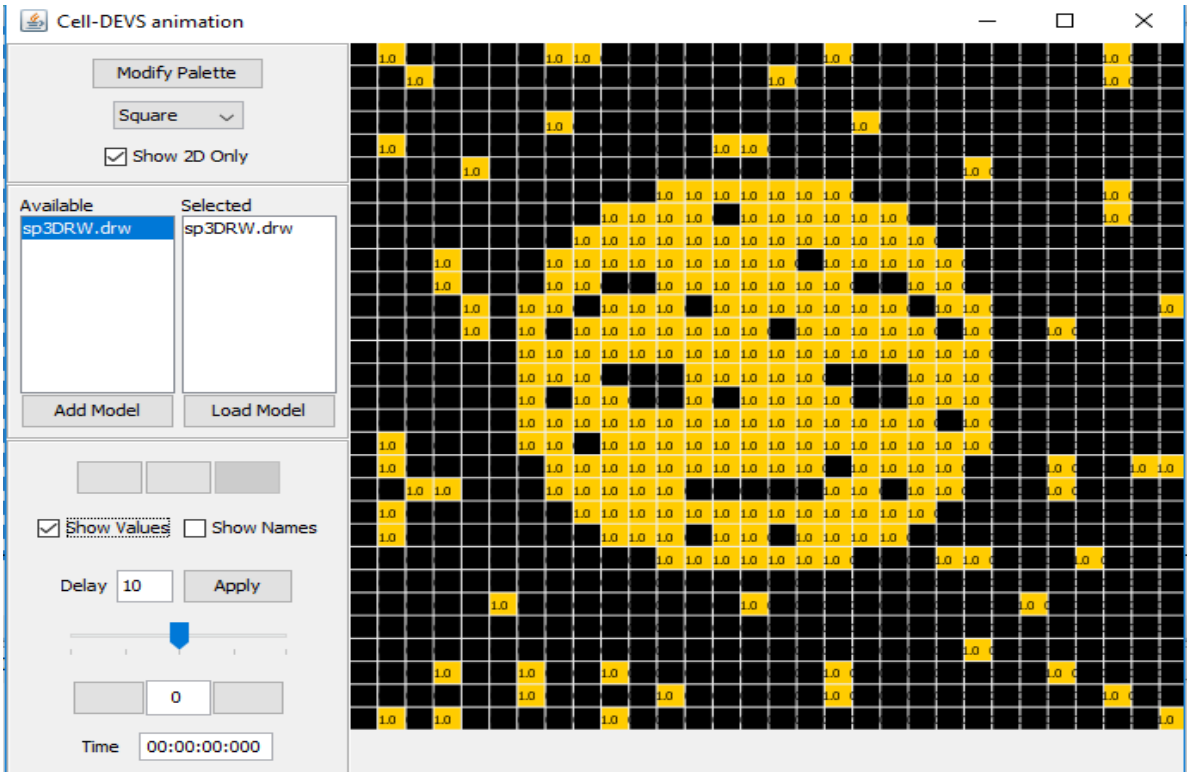


Figure 5 Smirking Emoji Image with Salt and Pepper Noise

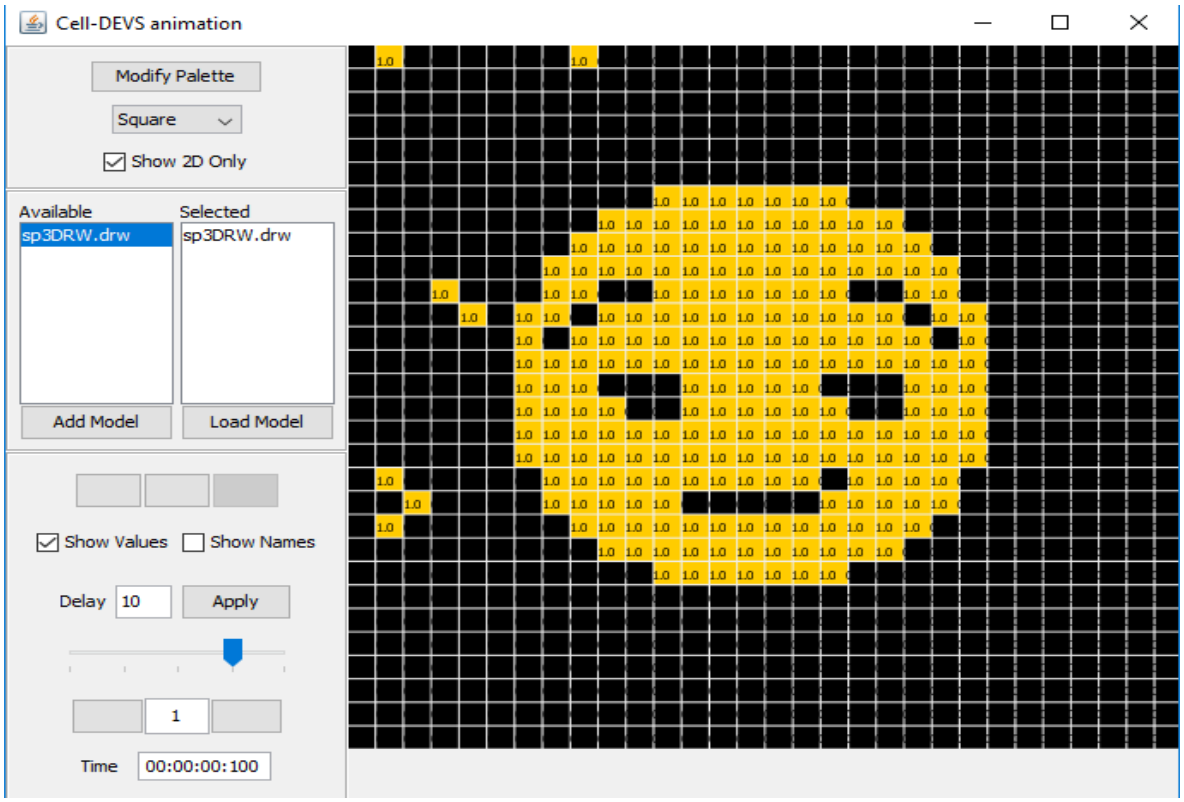


Figure 6 Noise Removed from the Object (Smirking Emoji) completely

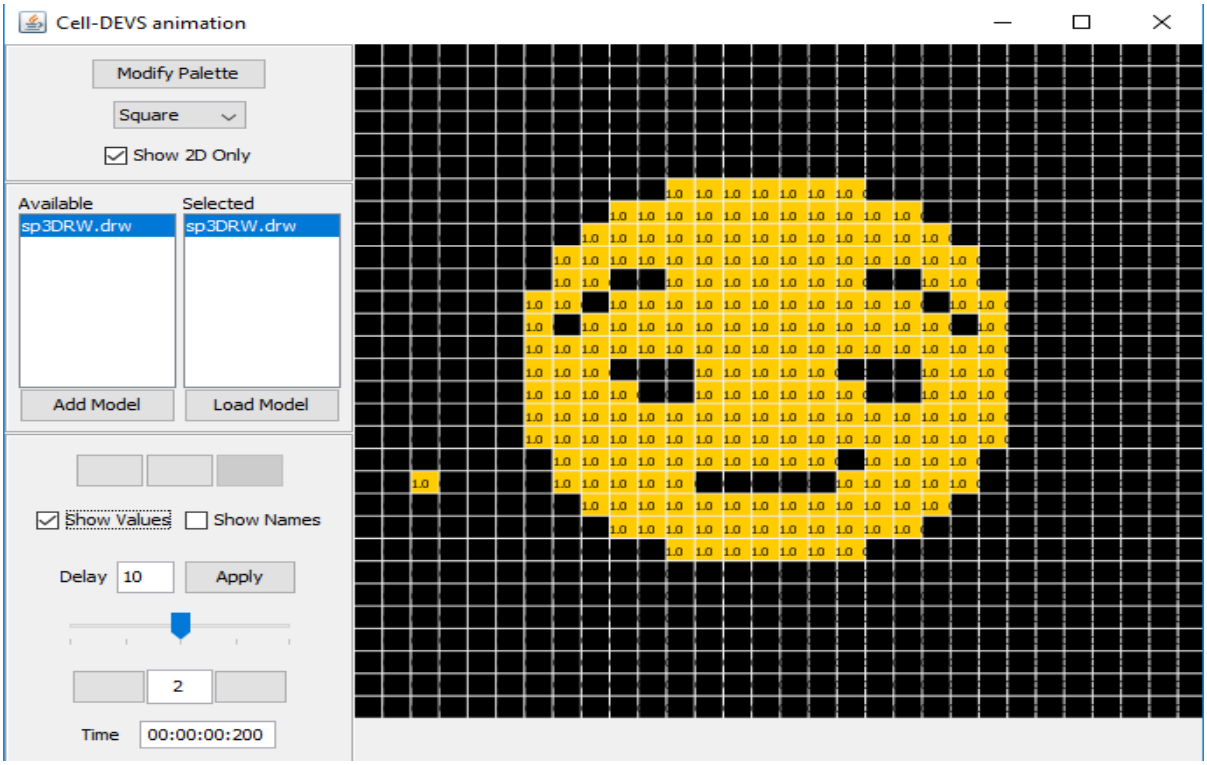


Figure 7 Noise is decreased in the background

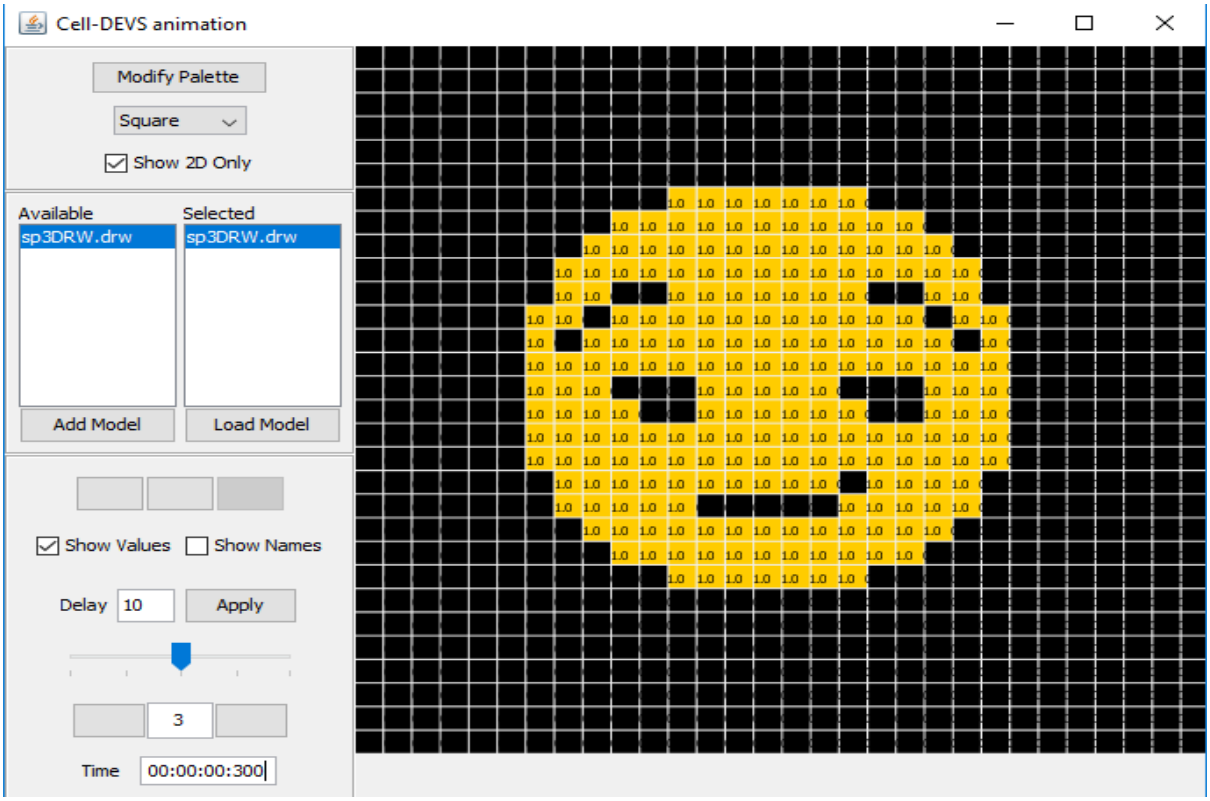


Figure 8 Noiseless Image as a Final Output

INTERESTING CONCLUSION:

Behavior of Salt and Pepper Noise filter is interesting in cellular automata and better than in classical approaches of image processing. In classical approaches, we analyze image pixels only once for noise removal and get results after one iteration. Here in cellular automata, we iterate this up till our rules of noise removal are not able to find any further noise in the image.

REFERENCES:

Anand Prakash Shukla, Suneeta Agarwal, *"Training Cellular Automata for Salt and Pepper Noise Filtering"*, International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity(CIPECH14) 28 & 29 November2014